

IC CARD AND METHOD FOR EXECUTING COMMAND

Patent Number: JP10040350
Publication date: 1998-02-13
Inventor(s): WAKAMATSU MASAKI; HARIMA HIROTSUGU
Applicant(s):: DAINIPPON PRINTING CO LTD
Requested Patent: ☐ JP10040350
Application Number: JP19960197318 19960726
Priority Number(s):
IPC Classification: G06K19/07
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To make it possible to use both of commands to be processed by a general program and an application program by selecting and executing the general program or the application program based on the judgement of a judging program and processing a command.
SOLUTION: At the time of receiving a command, an IC card 10 judges whether the command is a general command or a specific command. At the time of judging the command as a general command, the command is processed by executing a general program stored in a ROM 12, and at the time of judging the command as a specific command, the command is processed by executing an application program stored in an EEPROM 16. Consequently such a trouble that a command usable when the IC card 10 is driven by the general program can not be used when the application program is executed can be solved.

Data supplied from the **esp@cenet** database - 12

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10 - 40350

(43) 公開日 平成10年(1998)2月13日

(51) Int. Cl. °

G 0 6 K 19/07

識別記号

庁内整理番号

F I

G 0 6 K 19/00

技術表示箇所

N

審査請求 未請求 請求項の数 2

O L

(全 11 頁)

(21) 出願番号 特願平 8 - 197318

(22) 出願日 平成8年(1996)7月26日

(71) 出願人 000002897

大日本印刷株式会社

東京都新宿区市谷加賀町一丁目1番1号

(72) 発明者 若松 雅樹

東京都新宿区市谷加賀町一丁目1番1号 大
日本印刷株式会社内

(72) 発明者 針間 博嗣

東京都新宿区市谷加賀町一丁目1番1号 大
日本印刷株式会社内

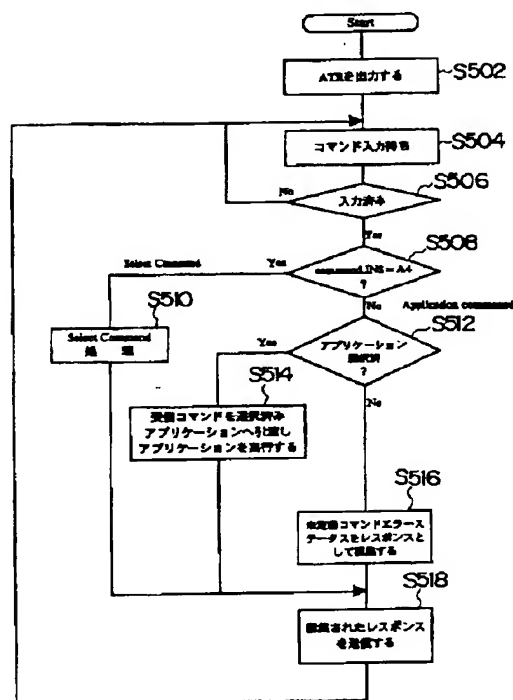
(74) 代理人 弁理士 鎌田 久男

(54) 【発明の名称】 ICカード及びコマンド実行方法

(57) 【要約】

【課題】 汎用プログラムにより処理されるコマンドと、アプリケーション・プログラムにより処理されるコマンドの双方を使用可能なICカードを提供する。

【解決手段】 CPUと、前記CPUが実行可能な汎用プログラムを格納する読み出し専用メモリと、前記CPUが実行可能なアプリケーション・プログラムを必要に応じて格納される書き換え可能な不揮発性メモリとを備えるICカードにおいて、前記読み出し専用メモリは、外部より付与されたコマンドが前記汎用プログラム又は前記アプリケーション・プログラムのいずれにより処理可能かを判断する判断プログラムを有し、前記判断プログラムの判断に基づいて、前記汎用プログラム又は前記アプリケーション・プログラムを選択して実行することにより前記コマンドを処理することを特徴とする。



【特許請求の範囲】

【請求項1】 CPUと、

前記CPUが実行可能な汎用プログラムを格納する読み出し専用メモリと、

前記CPUが実行可能なアプリケーション・プログラムを必要に応じて格納される書き換え可能な不揮発性メモリとを備えるICカードにおいて、

前記読み出し専用メモリは、外部より付与されたコマンドが前記汎用プログラム又は前記アプリケーション・プログラムのいずれにより処理可能かを判断する判断プログラムを有し、

前記判断プログラムの判断に基づいて、前記汎用プログラム又は前記アプリケーション・プログラムを選択して実行することにより前記コマンドを処理することを特徴とするICカード。

【請求項2】 ICカードのコマンド実行方法において、

外部より付与されたコマンドが、読み出し専用メモリに格納された汎用プログラム、又は不揮発性メモリに格納されたアプリケーション・プログラムのいずれにより処理可能かを判断し、

前記判断の結果に基づいて、前記汎用プログラム又は前記アプリケーション・プログラムを選択して実行することにより前記コマンドを処理することを特徴とするコマンド実行方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、不揮発性メモリにアプリケーション・プログラムを格納することが可能なICカードに関するものである。

【0002】

【従来の技術】ICカードは、磁気カードに代わる新しい情報記憶媒体として、近年注目を集めている。特に、CPUを内蔵したICカードは、高度なセキュリティを実現できることから、高度情報化社会の種々の分野において利用されることが期待されている。一般にICカードは、ROM、RAM、EEPROMの3種類のメモリと、それらメモリにアクセスするCPUとを備えている。EEPROMは、書き換え可能な不揮発性メモリであり、ICカードユーザに関する個人情報等のデータが保存される。RAMは、CPUがプログラムを実行するときに作業領域として使用する揮発性のメモリである。ROMは、読み出し専用メモリであり、CPUが実行すべき処理を示すプログラムが格納されている。

【0003】ICカードを使用するときは、ICカードをリーダー・ライターに接続し、リーダー・ライターからコマンドをICカードに送信する。コマンドを受信したICカードでは、CPUが、ROMに格納されているプログラムのうち、コマンドに対応する部分を実行する。この結果、コマンドの処理がなされ、EEPROMに新たなデ

ータを書き込むなどの処理が行われる。ROMが格納するプログラムは、ICカードの用途ごとに、つまり、実行すべきコマンドの種類や内容により異なる。したがって、ICカードの用途が多様化すると、それだけ多くの種類のROMを用意することとなる。しかし、一般にROMは、その開発に多大の費用を必要とするため、ICカードの用途の多様化に合わせ、ROMを多品種少量生産することとすれば、ROMの単価が上がり、ICカードの製造原価が増大する。

10 【0004】上記の問題に対し、従来ROMに格納されていたプログラムを2種類に大別し、一方をROMに、他方をEEPROMに格納するICカードが提案されている。ここで、2種類のプログラムの一方は、ICカードの用途によらず汎用的に使用できるプログラム（以下「汎用プログラム」という）である。また、他方は、ICカードの各種用途固有の処理を行うプログラム（以下「アプリケーション・プログラム」という）である。上記のICカードでは、全用途のICカードについて、同一のROMを使用することとなるので、安価なROMを用いた安価なICカードの生産が可能となる。一方、ICカードのEEPROMには、各用途に対応したアプリケーション・プログラムが書き込まれることから、用途に応じた本目細かい処理を行うICカードを提供することも可能となる。

【0005】

【発明が解決しようとする課題】しかし、前述した従来のICカードでは、一旦、アプリケーション・プログラムがEEPROMに導入された後は、ICカードはそのアプリケーション・プログラムにのみ従い動作する。したがって、アプリケーション・プログラム実行中は、そのアプリケーション・プログラムが処理できるコマンドのみが使用可能であり、汎用プログラムにその処理内容が定められているコマンドは使用することができなくなる。このために、汎用性のあるコマンドが、アプリケーション・プログラムの導入、利用により使用できなくなる場合があり、大変不便であるという問題があった。また、このような不都合を回避する一つの方法として、アプリケーション・プログラムにも、汎用的なコマンドに対応した処理内容を付加するという方法が考えられる。しかし、この場合には、アプリケーション・プログラムの規模が大きくなり、EEPROMのメモリ消費量が増大すると同時に、アプリケーション・プログラムと汎用プログラムとに重複する部分があるため、メモリの使用効率が著しく悪くなるという問題があった。

【0006】そこで、本発明の課題は、汎用プログラムにより処理されるコマンドと、アプリケーション・プログラムにより処理されるコマンドの双方を使用可能なICカードを提供することである。

【0007】

50 【課題を解決するための手段】前記課題を解決するため

に、請求項 1 に係る発明は、CPU と、前記 CPU が実行可能な汎用プログラムを格納する読み出し専用メモリと、前記 CPU が実行可能なアプリケーション・プログラムを必要に応じて格納される書き換え可能な不揮発性メモリとを備える IC カードにおいて、前記読み出し専用メモリは、外部より付与されたコマンドが前記汎用プログラム又は前記アプリケーション・プログラムのいずれにより処理可能かを判断する判断プログラムを有し、前記判断プログラムの判断に基づいて、前記汎用プログラム又は前記アプリケーション・プログラムを選択して実行することにより前記コマンドを処理することを特徴とする。

【０００８】請求項２に係る発明は、ＩＣカードのコマンド実行方法において、外部より付与されたコマンドが、読み出し専用メモリに格納された汎用プログラム、又は不揮発性メモリに格納されたアプリケーション・プログラムのいずれにより処理可能かを判断し、前記判断の結果に基づいて、前記汎用プログラム又は前記アプリケーション・プログラムを選択して実行することにより前記コマンドを処理することを特徴とする。

【 0 0 0 9 】

【発明の実施の形態】以下、図面等を参照して、本発明に係る一実施形態について、さらに詳しく説明する。図1は、本発明に係るＩＣカードの構成を示す図である。図1に示されるように、ＩＣカード10は、読み出し専用メモリであるROM12、揮発性メモリであるRAM14、随時書き換え可能な不揮発性メモリであるEEPROM16、及びこれらのメモリにアクセスするCPU18を備えている。

【0010】また、ICカード10は、リーダ・ライタ（下図示）と電気信号等の授受を行うための複数の接点（Vcc～GND）を備えている。ICカードをリーダ・ライタに挿入すると、リーダ・ライタの接点がこのICカードの接点と接続され、電気信号の授受が行われる。各接点には、それぞれ異なる電気信号が割り付けられている。例えば、Vccは、ICを動作させるために必要な電源電圧の供給を受けるための接点であり、I/Oは、CPUがリーダ・ライタとの通信を行うための接点（シリアルポート）である。

【0011】CPU18は、上記接点を介してコマンドを付与される。コマンドとは、リーダー/ライターからICカードへ送られる情報であって、ICカードに所定の動作させるためのものをいう。CPU18は、コマンドを付与されると、ROM12又はEEPROM16に格納されているプログラムを実行することによりそのコマンドを処理する。すなわち、CPU18は、EEPROM16に格納されているデータを書き換える等のコマンドに対応した所定の動作を行う。なお、本実施形態では、CPU18が実行すべきプログラムを前述した汎用プログラムとアプリケーション・プログラムとに分割し、汎

用プログラムをROM12に、アプリケーション・プログラムをEEPROM16に格納している。

【0012】図2は、本実施形態の各メモリに割り付けられているアドレスの一例を示す図である。本実施形態では、例えばROM12にH' 0000~H' 27FF、RAM14にH' 4000~H' 40FF、EEPROM16にH' 6000からH' 7FFFなるアドレスを割り当てている。なお、本明細書において数値の前に「H'」が付されているときは、その数値は16進数に基づいて標記されていることが意味される。

【0013】図3は、EEPROM16のメモリ・マップを示す図である。本実施形態では、EEPROM16の先頭の領域、すなわちH'6000からH'600Fをシステムエリアとして確保している。システムエリアの先頭アドレスH'6000には、変数NOAが格納されている。NOAは、EEPROM16に格納（登録）されているアプリケーション・プログラムの個数を示す変数である。NOAは、0に初期設定され、後述するように1のアプリケーション・プログラムがEEPROM16に格納されるごとにその値を1ずつ加算される。アドレスH'6001～H'600Fまでの領域（RFU）は、将来ICカードの使用を変更したときに、その仕様において必要とされる各種パラメータ等を格納するための予備的な空白領域である。

【0014】アドレスH'6010以降は、アプリケーション・プログラムを格納することが可能な領域である。図3には、一例として、2つのアプリケーション・プログラムが格納されている状態が示されている。第1のアプリケーション・プログラムはH'6010~H'607Fの領域に、第2のアプリケーション・プログラムは、H'6080~H'60DBの領域に格納されている。EEPROM16に格納されているアプリケーション・プログラムは、an_1からapplication program (以下「ap. pro.」と略す)までの5つの情報から構成されている。an_1及びapplication_name (以下「ap. na.」と略す)は、アプリケーション・プログラムを識別するための情報である。すなわち、ap. na. は、当該アプリケーション・プログラムの名前であり、an_1は、ap. na. の長さ(バイト数)を示す1バイトのデータである。

【0015】NA__ADDは、当該アプリケーション・プログラムの次に格納されている、又は格納されるべきアプリケーション・プログラムの先頭アドレスを示す2バイトのデータである。図3の例では、第1のアプリケーション・プログラムが有するNA__ADDは、第2番目のアプリケーション・プログラムの先頭アドレスH'6080を示している。また、第2番目のアプリケーション・プログラムが有するNA__ADDは、将来第3番目のアプリケーション・プログラムが格納されるべき領

域の先頭アドレスH'60DCを示している。なお、本実施形態では、NA_ADDは、必ずap_na.の次の領域に格納される。

【0016】apl_1は、NA_ADDの次に格納される2バイトのデータであり、その直後に格納されているap_pro.の長さ(バイト数)を示している。ap_pro.は、アプリケーション・プログラムの処理内容を示すプログラムと、そのプログラム実行時に参照されるデータの集合である。図8は、ap_pro.の構成を示す模式図である。ap_pro.は、プログラム領域とデータ領域とから構成されている。プログラム領域には、そのアプリケーション・プログラムに固有なコマンドの処理内容等が格納されている。また、データ領域には、アプリケーション・プログラムが取り扱うデータが格納されている。例えば、アプリケーション・プログラムがA銀行の預貯金に関するプログラムである場合には、A銀行におけるICカード所有者の預金残高等がデータ領域に格納される。

【0017】図4は、本実施形態で使用されるコマンドの一例を示す図である。本実施形態で使用されるコマンドは、汎用コマンドと特定コマンドの2種類に大別される。汎用コマンドとは、ICカードの使用態様によらず、一般的に使用され得るコマンドをいう。汎用コマンドを付与された場合に、CPU18が処理すべき内容は、ROM12に格納されている汎用プログラムに定められている。一方、特定コマンドとは、ICカードの特定の使用態様に固有な処理を行うためのコマンドである。例えば、アプリケーション・プログラムのデータ領域にデータを書き込むためのコマンド、又は、データを読み出すためのコマンドは特定コマンドとなり得る。アプリケーション・プログラムにより、データ領域におけるデータのフォーマットが相違することがあるからである。特定コマンドに対応する処理内容は、アプリケーション・プログラムのプログラム領域に定められている。

【0018】図4に示すコマンドのうち、Selectコマンドは汎用コマンドであり、WriteコマンドとRead_recordコマンド(以下「Readコマンド」と略す)は特定コマンドである。Selectコマンドは、EEPROM16に格納されているアプリケーション・プログラムから任意の1つを選択するためのコマンドである。Selectコマンドは、3つの情報より構成されている。第1バイト目のINSは、コマンドを識別するための種別コードである。ここでは、INSに例えば「H'A4」を割り当てている。また、第2バイト目はan_1、第3バイト目以降は、ap_na.である。

【0019】Writeコマンドは、指定するデータをアプリケーション・プログラムのデータ領域に書き込むためのコマンドである。Writeコマンドは、第1バイト目から順にINS、W_len及びN_dataの

3つの情報から構成されている。N_dataは、データ領域に書き込まれるべきデータである。また、W_lenは、N_dataの長さ(バイト数)を示す1バイトのデータである。なお、INSとしては、例えば「H'B0」を割り当てている。Readコマンドは、アプリケーション・プログラムのデータ領域から指定する長さのデータを読み出すためのコマンドである。Readコマンドは、INSとRead_lenの2つの情報から構成される。Read_lenは、読み出すべきデータの長さ(バイト数)を示す1バイトデータである。また、INSには、例えば「H'C0」を割り当てている。

【0020】上記のように、本実施形態では、汎用コマンド及び特定コマンドの2種類のコマンドがあるために、リーダー・ライタからコマンドが受信された場合に、ICカード10は、まずコマンドが汎用コマンドであるか、特定コマンドであるかを判別し、その判別結果に応じたコマンド処理を行う。以下、本実施形態におけるコマンドの処理方法について、図5等を用いて説明する。

【0021】図5は、ROM12に格納されている汎用プログラムの流れ図である。ICカード10とリーダー・ライタとが接続され、リーダー・ライタによりICカードがリセット(活性化)されると、CPU18は、まず初期応答情報(ATR)をレスポンスとして出力する(S502)。次に、CPU18は、リーダー・ライタからのコマンド待ち状態となる(S504、S506)。リーダー・ライタからのコマンドを受信すると(S506:Yes)、CPU18は、そのコマンドのINSが汎用コマンドのものであるか否かを確認する(S508)。例えば図示の例では、INSが「A4」であるか否かが確認される。INSが「A4」であると、コマンドがSelectコマンド(汎用コマンド)であると判断され、ROM12に格納されている汎用プログラムを実行することによりSelectコマンドの処理がなされる(S510)。

【0022】一方、S508において、INSの内容が汎用コマンドのINSに対応しなかった場合は、コマンドが特定コマンドであると判断され、S512の処理へ移行する。S512では、既にSelectコマンドが処理され、所定のアプリケーション・プログラムが選択済みであるか否かが判断される。具体的には、RAM14の所定領域に、Selectコマンドを処理することにより特定された、アプリケーション・プログラムのアドレスが存在するか否かが判断される。S512において、アプリケーション・プログラムが選択済みであると判断されると、受信されたコマンドのパラメータに基づいて、当該アプリケーション・プログラムが実行される(S514)。

【0023】一方、S512において、アプリケーションが選択されていないと判断されると、コメント・エラー

一・ステータスがRAM14の所定領域にレスポンス情報として編集される(S516)。次に、S510、S514又はS516のいずれかの処理が終了すると、RAM14の所定領域に編集されたレスポンス情報が、リーダー・ライタに送信される(S518)。レスポンス情報が送信された後は、再びS504に戻り、S518までの処理が繰り返される。

【0024】図6は、S510の処理内容、つまり、Selectコマンドが受信された場合にCPU18が実行する処理内容を示す流れ図である。S510において、CPU18は、はじめに変数search_address(以下「s. a.」と略す)をアプリケーション・プログラムが格納されている領域の先頭アドレスに初期設定する(S602)。本実施形態の場合には、s. a. は、H'6010に設定される。次に、CPU18は、EEPROM16のシステムエリアよりNOAの値を読み出し(S604)、その値をもって変数counterを設定する。例えば図3に示すようにEEPROM16が2つのアプリケーション・プログラムを格納している場合には、counterは2に初期設定される。

【0025】次に、CPU18は、Selectコマンドに指定されたアプリケーション・プログラムと同一のものがあるか否かについて、EEPROM16に格納されているアプリケーション・プログラムを順次検索する(S608~S616)。具体的には、s. a. が示すアドレスのan_1及びそのan_1に続くap. n a. と、Selectコマンドのan_1及びap. n a. とが比較される(S610)。比較の結果、一致しない場合には、counterの値を1だけデクリメントするとともに、s. a. にNA_ADDの内容を代入する(S614、S616)。ここで、S616におけるNA_ADDは、S610においてs. a. が示したアドレス以降に現れる最初のNA_ADDである。S616においてs. a. の内容を更新することにより、s. a. は、次に格納されているアプリケーション・プログラムの先頭アドレスを示すこととなる。

【0026】S610からS616までの処理は、S612において比較されたデータが一致するまで、又は、S608においてcounterの値が0となるまで継続される。S612において、比較されたデータが一致した場合(S612:Yes)には、Selectコマンドによって指定されたアプリケーション・プログラムが発見され、s. a. はその先頭アドレスを示していることが意味される。そこで、CPU18は、s. a. にan_1の長さ(バイト数)及びan_1が示す値、すなわちap. n a. の長さ(バイト数)、NA_ADDの長さ、さらにはap_1_1の長さを順次加算し、これを変数s. a. に代入する(S618)。この結果、s. a. は、当該アプリケーション・プログラムにお

るap. pro. の先頭アドレスを示すこととなる。CPU18は、このs. a. の内容をRAM14上の所定領域に格納する(S620)。そして、Selectコマンドの処理が正常に終了した旨のステータスをRAM14の所定領域にレスポンス情報として格納し(S622)、処理を終了する。

【0027】一方、S608において、counterの値が0となった場合(S608:Yes)には、EEPROM16に格納されているいずれのアプリケーション・プログラムもSelectコマンドが指定するものに該当しなかったことが意味される。この場合には、CPU18は、該当するアプリケーション・プログラムがなかった旨のエラーステータスをRAM14の所定領域にレスポンス情報として格納し(S624)、処理を終了する。

【0028】図7は、図5におけるS514において、CPU18が実行する処理内容を示した流れ図である。図5のS512において、アプリケーションが選択済みであると判断されると、CPU18は、図6のS622においてRAM14に格納されたs. a. の内容を取得し、それに示されるアドレスをサブルーチンコールする(S702)。これにより、そのアプリケーション・プログラムの実行が開始され、受信されたコマンドの処理が行なわれる。例えば、受信されたコマンドがWriteコマンドである場合には、アプリケーション・プログラムのプログラム領域のうち、Writeコマンドの処理に対応する部分(図8参照)等が実行される。この結果、アプリケーション・プログラムのデータ領域には、WriteコマンドのN_dataの内容が書き込まれる。また、受信されたコマンドがReadコマンドである場合には、プログラム領域のうち、Readコマンドの処理に対応する部分等が実行され、データ領域の先頭からRead_lenバイトの情報がRAM14の所定領域にレスポンス情報として編集される。

【0029】以上説明したように、本実施形態では、ICカードがコマンドを受信すると、はじめにそのコマンドが汎用コマンドであるのか、特定コマンドであるのか判断される。そして、汎用コマンドであると判断された場合には、ROM12に格納されている汎用プログラムを実行することによりそのコマンドの処理を行い、一方、特定コマンドであると判断された場合には、EEPROM16に格納されているアプリケーション・プログラムを実行することによりそのコマンドの処理を行う。

【0030】これにより、本実施形態では、汎用プログラムに従いICカードを動作させていたときには使用可能であったコマンドが、アプリケーション・プログラムを導入・実行したために、使用できなくなるという従来の不都合が消される。また、その結果として、アプリケーション・プログラムは、汎用プログラムと重複したコマンドの処理内容を備える必要がなくなり、従来よりコ

ンパケットなものとなる。したがって、本実施形態は、アプリケーション・プログラムの導入によるEEPROMのメモリ消費量が最小限に抑制され、ICカードの限られたメモリ資源を有効に活用することが可能となっている。

【0031】

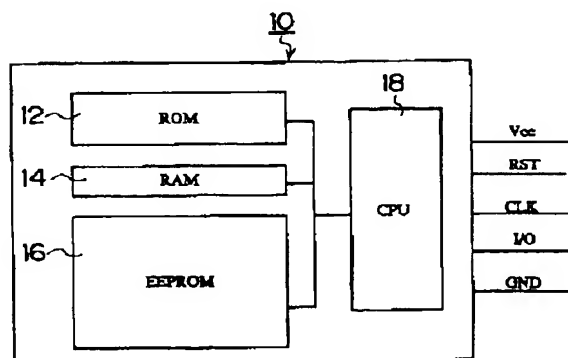
【発明の効果】以上、詳しく説明したように、本発明によれば、アプリケーション・プログラムを導入した後に、汎用プログラムにより処理されるコマンドと、アプリケーション・プログラムにより処理されるコマンドの双方を使用可能なICカードを提供することが可能となった。

【図面の簡単な説明】

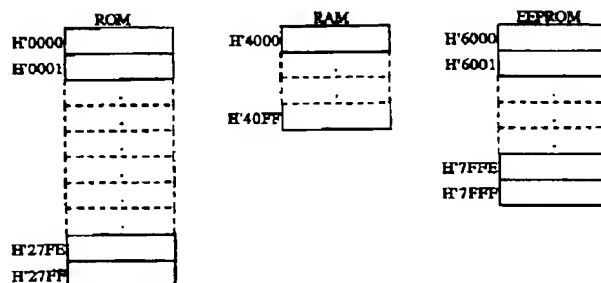
【図1】本発明に係るICカードの構成を示す図である。

【図2】本発明に係るICカードの各メモリに割り付けられているアドレスの一例を示す図である。

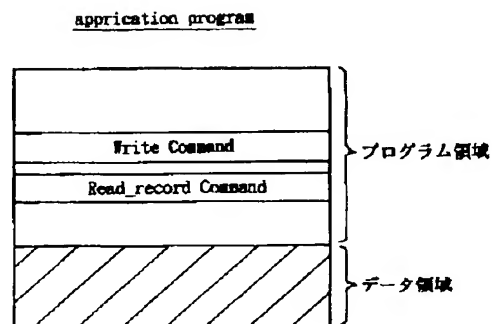
【図1】



【図2】



【図8】



【図3】EEPROM16のメモリ・マップである。

【図4】本発明の実施形態で使用されるコマンドの一例を示す図である。

【図5】ROM12に格納されている汎用プログラムの流れ図である。

【図6】Selectコマンドが受信された場合にCPU18が実行する処理内容を示す流れ図である。

【図7】図5のステップ514において、CPU18が実行する処理内容を示す流れ図である。

10 【図8】application programの構成を示す模式図である。

【符号の説明】

- 10 ICカード
- 12 ROM
- 14 RAM
- 16 EEPROM
- 18 CPU

【図3】

EEPROM MAP

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
H'6000	NOA	RPU														
H'6010	an_l	application_name													NA_ADD	
H'6020	apl_l															
H'6030																
H'6040																
H'6050																
H'6060																
H'6070																
H'6080	an_l															
H'6090																
H'60A0																
H'60B0																
H'60C0																
H'60D0																
H'7FF0																

EEPROM : = System Area + Application Loading Area
 (System Area - H'6000 .. H'600F
 Application Loading Area - H'6010 .. H'7FFF)

NOA : EEPROMに登録されているアプリケーションの個数
 an_l : Application name length
 application name :
 NA_ADD : Next Application ADDRESS
 apl_l : application programの長さ
 application program : アプリケーションプログラムCPU命令コード

* : EEPROMの初期値は、ALL H'00 とする。

【図4】

Select Command フォーマット

INS	an_l	application_name
(1)	(1)	(an_l)

INS : Application Load Command種別コード (H'A4)
 an_l : application_name の長さ
 application_name : アプリケーションの名前

Application Command フォーマット

INSが H'A4 以外のコマンドをApplication Commandとする

Write Command フォーマット

INS	W_len	N_data
(1)	(1)	(W_len)

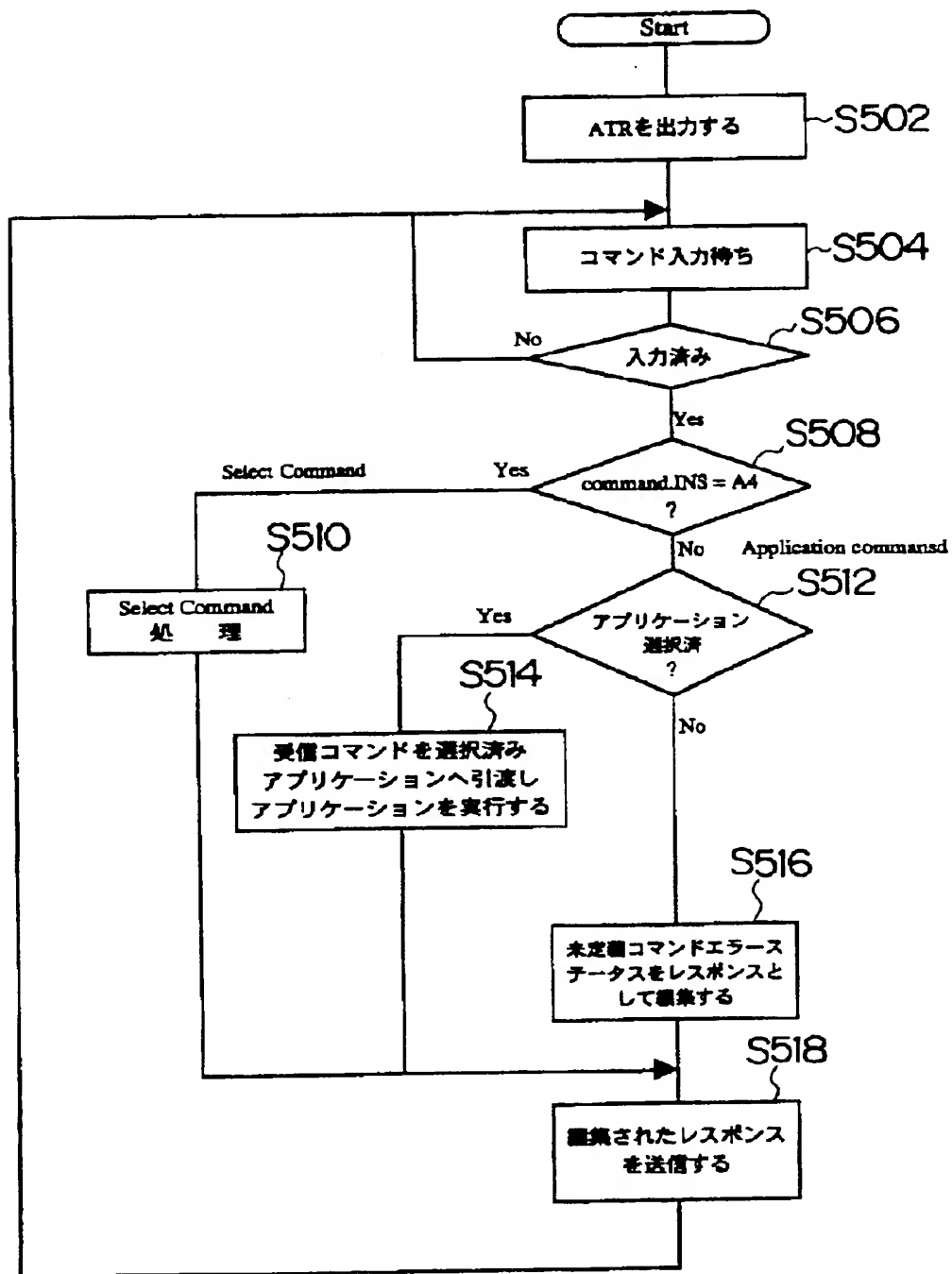
INS : (H' B0)
 W_len : N_dataの長さ
 N_data:書き込むべきデータ

Read_record Command フォーマット

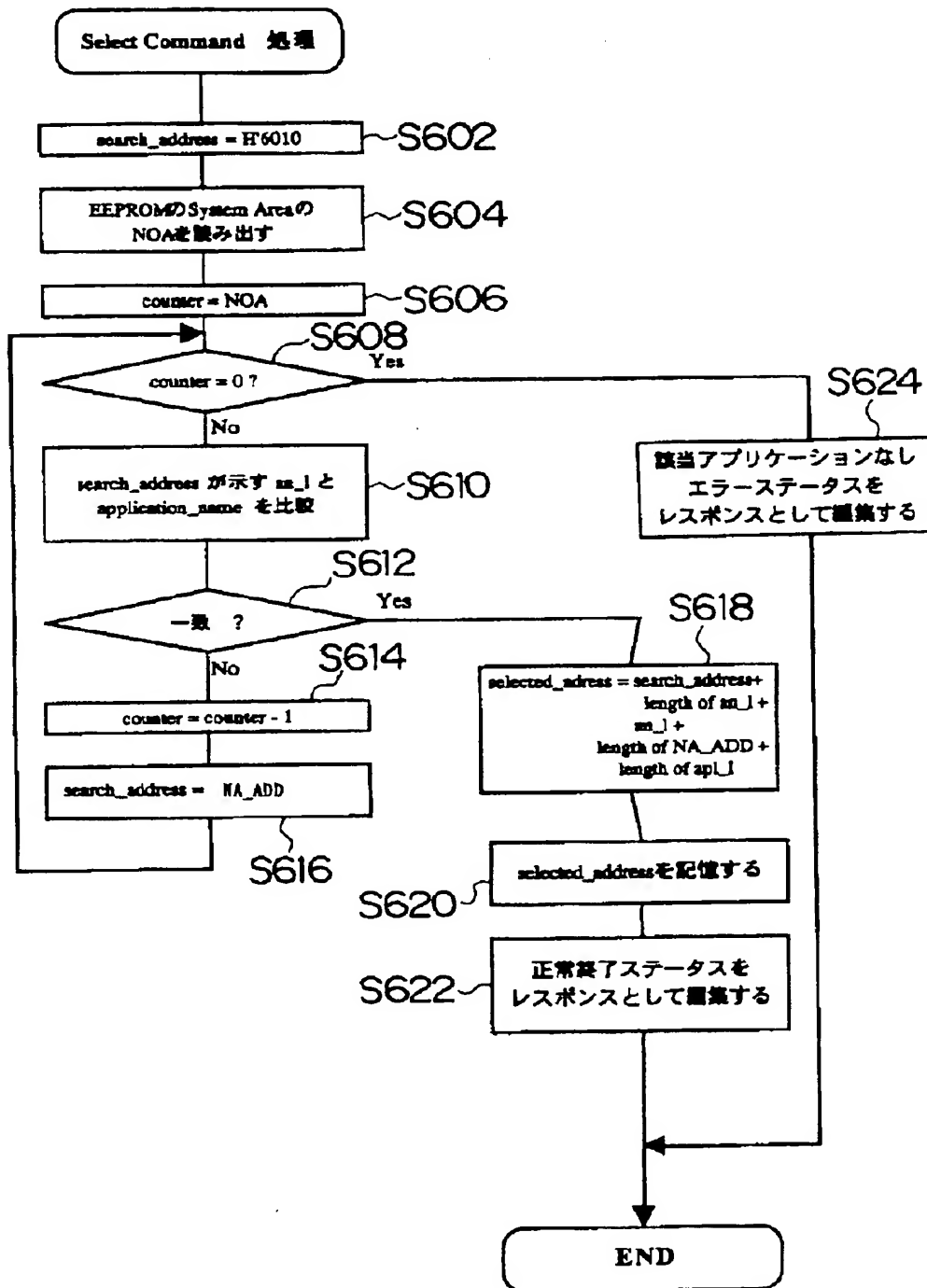
INS	Read_len
(1)	(n)

INS : (H' C0)
 Read_len:読み取るデータの長さ

【図5】



【図6】



【図7】

